



(11) **EP 1 811 373 A1**

(12) **DEMANDE DE BREVET EUROPEEN**

(43) Date de publication:
25.07.2007 Bulletin 2007/30

(51) Int Cl.:
G06F 9/44 (2006.01)

(21) Numéro de dépôt: **06300062.4**

(22) Date de dépôt: **24.01.2006**

(84) Etats contractants désignés:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI SK TR
Etats d'extension désignés:
AL BA HR MK YU

(74) Mandataire: **Sciaux, Edmond**
Alcatel Lucent
Intellectual Property & Standards
54 rue La Boétie
75008 Paris (FR)

(71) Demandeur: **Alcatel Lucent**
75008 Paris (FR)

Remarques:
Revendications modifiées conformément à la règle 86 (2) CBE.

(72) Inventeur: **LARVET, Philippe**
91470, FORGES LES BAINS (FR)

(54) **Procédé de composition automatique de services web, produit de programme d'ordinateur et système informatique de mise en oeuvre de ce procédé**

(57) L'invention concerne la composition, par exemple l'orchestration ou la chorégraphie de services web.

L'invention propose un procédé de composition automatique de services web. Ce procédé comprend une étape d'identification de caractéristiques de services web, à partir d'une description standardisée de services web. Recourir à une description standardisée permet de

systematiser le procédé. Une fois identifiées, ces caractéristiques permettent de déterminer une connectivité de ces services, c'est-à-dire la propension qu'ont ces services à se connecter. Ensuite, il est possible de composer automatiquement ces services en fonction de leur connectivité.

EP 1 811 373 A1

Description

[0001] La présente invention concerne la composition de services web, par exemple l'orchestration ou la chorégraphie de tels services.

5 **[0002]** Dans la présente description, les termes suivants sont employés avec le sens indiqué, sauf mention particulière :

- « annuaire UDDI », pour « Universal Description Discovery and Integration », désigne un annuaire basé sur XML, dédié plus particulièrement aux services web, notamment dans le cadre d'architectures de type SOA (pour « Service Oriented Architecture »). Il existe des annuaires privés et publics, des annuaires spécifiques pour certains fournisseurs de services. Un annuaire UDDI permet de localiser sur le réseau un service Web recherché. Il comprend notamment des informations d'accès aux services Web et des informations contextuelles, par exemple une brève description des services et leurs fonctionnalités. L'annuaire UDDI contient notamment :
 - des pages blanches comprenant des informations relatives à des entreprises ;
 - des pages jaunes, qui recensent des services web de ces entreprises au standard WSDL ; et
 - des pages vertes fournissant des informations techniques précises sur les services fournis. Ces informations techniques concernent entre autres des descriptions de services, de liaison ou de processus métiers associés.
- « application informatique » : désigne tout programme ou tout ensemble de programmes prévus pour fonctionner conjointement, dont le but est de rendre des services à leurs utilisateurs.
- « BPEL » : Le langage BPEL, pour « Business Process Execution Language », est un langage XML utilisé pour la notation formelle de logiques d'interactions et de protocoles d'interactions, qui étend la sémantique du modèle d'interaction de services Web. Au-delà des interactions entre web services, BPEL décrit la logique applicative et l'ordre des interactions.
- « composition » : dans le contexte des services Web, désigne une agrégation de services susceptibles d'agir comme un nouveau service à part entière. On distinguera la composition par "orchestration", dans laquelle l'accent est mis sur l'ordonnancement des traitements effectués par les services composés, et la composition par "chorégraphie", dans laquelle l'ordonnancement des services se base sur les messages échangés et sur la prise en compte des événements par les services composés.
- « paramètre d'entrée » désigne une donnée fournie en entrée à une opération de traitement d'un service web ; les données d'entrée de l'opération sont transformées par l'opération pour fournir le résultat de sortie.
- « résultat de sortie » désigne l'information résultant du traitement effectué par une opération d'un service web sur ses données (ou "paramètres") d'entrée. Le résultat de sortie est le produit attendu par l'utilisateur du service web, cet utilisateur pouvant être une application informatique ou un autre service web.
- « service Web » désigne une application accessible sur l'INTERNET, via une interface standard, qui peut interagir dynamiquement avec des applications ou d'autres services web en utilisant des protocoles de communication, par exemple basés sur le XML, et ce, indépendamment du système d'exploitation et des langages de programmation utilisés. Au plan de ses interfaces proprement dites, un service web comprend des opérations de traitement qui fournissent des résultats à partir de données d'entrée, ou "paramètres d'entrée". Pour utiliser un service web, on appelle une de ses opérations en lui fournissant les données d'entrée attendues, et on récupère le résultat de sortie.
- « SGML » : un langage normalisé permettant de décrire les relations entre le contenu d'un document informatique et sa structure.
- « WSDL » est un langage normalisé par le W3C (version 1.1 du 15 mars 2001 - voir <http://www.w3.org/TR/wsdl>), faisant notamment référence au langage XML. Le WSDL décrit une Interface publique d'accès à un service web par les éléments suivants : nom du service, opérations, types de données manipulées et liens entre ces types et les opérations. Le WSDL indique ainsi comment communiquer pour utiliser le service, le protocole de communication et le format de messages requis pour communiquer avec ce service.
- « XML » (pour extensible Markup Language) : une évolution du langage SGML, laquelle permet notamment aux concepteurs de documents HTML de définir leurs propres marqueurs, dans le but de personnaliser la structure des données.

[0003] En programmation, il est souvent nécessaire de composer des services web. La solution courante à ce problème est de composer « manuellement » ces services. En particulier, il est nécessaire de recourir à un langage de programmation, par exemple Java, C# ou un langage de composition de service web spécifique tel que BPEL (pour « Business Process Execution Language »).

[0004] Cette solution n'est cependant pas satisfaisante parce qu'elle est manuelle au sens où elle implique de la programmation. Elle est donc fastidieuse, lente et susceptible d'occasionner des erreurs.

[0005] Il existe donc un besoin pour un procédé de composition automatique de services web.

[0006] A cette fin, l'invention propose un procédé de composition automatique de services web, comprenant les étapes : d'identification de caractéristiques de services web à partir d'une description standardisée de services web ; de détermination d'une connectivité de ces services en fonction d'au moins une caractéristique identifiée pour chacun des services ; et de composition de ces services en fonction de leur connectivité.

[0007] Dans des modes de réalisation préférés, le procédé de composition automatique de services web selon l'invention comprend une ou plusieurs des caractéristiques suivantes :

- les caractéristiques identifiées comprennent des étiquettes sémantiques ;
- l'étape d'identification de caractéristiques de services web comprend, pour chacun des services : l'accession à une description standardisée du service web ; et l'identification de caractéristiques de ce service à partir la description standardisée de ce service ;
- chaque description est en langage WSDL ;
- l'étape d'identification de caractéristiques comprend une analyse syntaxique, et sémantique de la description de services web ;
- les caractéristiques identifiées impliquent au moins une entrée et/ou une sortie pour chacun des services ;
- l'étape de détermination de la connectivité des services web comprend : une étape de test de compatibilité impliquant au moins une caractéristique identifiée pour chacun des services ;
- les caractéristiques comprennent, pour chacun des services web : au moins deux ensembles comprenant au moins trois champs chacun, un premier champ de l'un des ensembles ayant trait à une entrée du service ; un premier champ de l'autre ensemble ayant trait à une sortie du service ; un deuxième champ de chacun des ensembles ayant trait à un type de données relatif au premier champ ; un troisième champ de chacun des ensembles ayant trait à une étiquette sémantique relative au premier champ, l'étape de test de compatibilité impliquant les trois champs de chacun des ensembles ;
- l'étape de composition comprend une étape d'orchestration des services web ; et
- l'étape de composition comprend une étape de chorégraphie des services web.

[0008] L'invention concerne en outre un procédé de création d'un nouveau service web comprenant les étapes du procédé de composition automatique de services web selon l'invention et comprenant en outre une étape de création d'un nouveau service web à partir de la composition des services web.

[0009] L'invention concerne également un produit de programme d'ordinateur adapté à mettre en oeuvre les étapes de l'un des procédés selon l'invention.

[0010] L'invention concerne encore un système informatique comprenant des moyens de code adaptés à mettre en oeuvre les étapes de l'un des procédés selon l'invention.

[0011] D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture de la description détaillée qui suit des modes de réalisation de l'invention, donnés à titre d'exemple uniquement.

[0012] L'invention propose un procédé de composition automatique de services web. Ce procédé comprend une étape d'identification de caractéristiques de services web, à partir d'une description standardisée de services web. Recourir à une description standardisée permet de systématiser le procédé. Une fois identifiées, ces caractéristiques permettent de déterminer une connectivité de ces services, c'est-à-dire la propension qu'ont ses services à se connecter. Ensuite, il est possible de composer automatiquement ces services en fonction de leur connectivité. Le procédé selon l'invention permet, pour deux ou plusieurs services web donnés, de composer automatiquement ces services. On part ici du principe que les services en question sont connus et que l'on souhaite effectuer une composition de ces services. Comme défini plus haut, le terme composition s'entend ici comme une généralisation des termes orchestration et chorégraphie.

[0013] Ce procédé est de préférence implémenté par un outil informatique « intégré », par exemple un assistant comprenant plusieurs modules couplés, chacun de ces modules étant assigné à une tâche donnée.

[0014] Une étape du procédé consiste à identifier des caractéristiques spécifiques de ces services web. Pour cela, il est fait appel à une description standardisée de ces services.

[0015] De préférence, on fait appel à des descriptions individualisées de services web. De la sorte, pour un service web donné, l'outil informatique accède à une description de ce service puis procède à l'identification de caractéristiques spécifiques à partir de sa description. Il convient de noter que ces caractéristiques n'existent pas en tant que telles dans la description du service, du moins pas explicitement. Elles sont déduites par le procédé de l'invention.

[0016] Il est préférable d'utiliser des descriptions respectives de services web, surtout si l'on dispose d'un catalogue de description, par exemple un annuaire UDDI, tel que défini plus haut. L'accession à de telles descriptions individuelles de services web revêt alors un caractère systématique : pour chaque service web impliqué, le procédé recherche une description correspondante.

[0017] De préférence, chaque description de services est en langage WSDL, comme c'est par exemple le cas au sein d'annuaires UDDI. On dispose ainsi d'une description standardisée pour chaque service web. En outre, une telle description est toujours disponible sur un serveur d'annuaire car elle est automatiquement construite par ce serveur à partir

de l'interface externe de la description. Ceci offre davantage de compatibilité au procédé.

[0018] Par ailleurs, il est possible d'analyser de manière automatique de telles descriptions, par exemple au moyen d'un analyseur syntaxique (ou « parser », c'est-à-dire un outil apte à analyser une description à partir de la reconnaissance d'unités syntaxiques) ou une routine d'analyse syntaxique. Il sera ci-après fait référence à un module sémantique plutôt qu'à un analyseur, le module étant, par définition, susceptible de s'intégrer dans un outil informatique, aux fonctions plus étendues. Le module a donc pour fonction de procéder à une analyse de la description WSDL d'un point de vue sémantique.

[0019] Plus en détails, l'analyse syntaxique comprend par exemple une analyse lexicale, laquelle permet de séparer des chaînes de symboles en éléments utilisables. Il est ensuite procédé à une analyse sémantique de la description de services web, afin de déterminer la signification des chaînes de caractères. Des outils d'analyse syntaxique sont connus et disponibles.

[0020] L'analyse de la description permet d'identifier des caractéristiques pour chacun des services. Par exemple, il permet d'identifier des ensembles de caractéristiques pour chacune des opérations des services. Ces caractéristiques sont alors utilisées pour déterminer, selon le procédé de l'invention, la connectivité des services étudiés.

[0021] Parmi les caractéristiques de services web, on cherche plus spécialement à identifier au moins une entrée et/ou une sortie pour chacune des opérations d'un service web donné.

[0022] Plus généralement, on peut par exemple rechercher les entrées-sorties de chaque opération de chaque service, en tant que "paramètres d'entrée" ou "résultats de sortie" (voir plus haut les définitions de ces termes). De la sorte, il sera ensuite possible de tester la connectivité des services web en comparant les caractéristiques d'entrées/sorties de leurs opérations. Par exemple, un service A ayant une opération fournissant une sortie compatible avec une entrée d'une opération d'un service B est susceptible d'être connecté avec ce dernier.

[0023] Afin de déterminer la connectivité des services et en vue de la composition future de ces services, on pourra, dans un mode de réalisation, constituer un ensemble de caractéristiques pour chaque entrée et chaque sortie des opérations d'un service web donné..

[0024] Si une opération d'un service web présente plusieurs entrées ou plusieurs sorties, un nombre correspondant d'ensembles pourra être établi lors de l'analyse de la description du service web.

[0025] En plus du nom du paramètre lui-même, ces ensembles peuvent par exemple présenter trois champs chacun (ou plus, selon le modèle de test envisagé). Notons cependant que l'utilisation de deux champs pourrait s'avérer suffisante selon l'interprétation qui est faite de la description WSDL.

[0026] Chaque ensemble peut donc, de préférence, présenter trois champs et se rapporter à un paramètre d'entrée ou à un résultat de sortie d'une opération du service web. Un premier champ se rapporte à l'information d'entrée ou de sortie pour le paramètre ou le résultat concerné. Un deuxième champ de chaque ensemble précise le type de données du paramètre ou du résultat. Le type de données est l'un des éléments présents dans la description WSDL du service. Un troisième champ se rapporte à une étiquette sémantique (ou « semantic tag » ou « SemTag » dans l'exemple plus loin), associée au paramètre ou au résultat. Une étiquette sémantique peut ici être vue comme un nom, un groupe de caractères ou plus généralement un code associé à la donnée et servant à la caractériser davantage au plan sémantique. Notons que si le type peut être directement extrait d'une description WSDL, l'étiquette est, elle, apposée après analyse de cette même description.

[0027] Le test de compatibilité des caractéristiques peut alors impliquer chaque champ de chaque ensemble.

[0028] On pourra faire appel au module sémantique, évoqué plus haut et qui fait typiquement partie intégrante de l'outil informatique précité, pour positionner une étiquette sémantique sur chaque type de données en entrée et en sortie. Cette étiquette précise la nature d'une donnée d'entrée/sortie et précise donc la manière dont la donnée peut être traitée. Chaque donnée d'entrée/sortie se trouve ainsi décrite par un triplet :

{entrée ou sortie, type de donnée, étiquette sémantique}.

[0029] On montre, ci-après, un extrait de fichier XML correspondant à une description sémantique, à l'aide de trois champs, des entrées d'un service web de traduction qui sera repris plus loin (cet extrait de fichier XML est issu d'un traitement sémantique effectué à partir d'un fichier WSDL correspondant) :

```

...
  <input>
5       <param name="src_lang" type="string" semtag="language" />
        <param name="dest_lang" type="string" semtag="language" />
        <param name="text_to_tr" type="string" semtag="text" />
10      </input>
...

```

15 **[0030]** On voit ainsi que chaque paramètre, en plus de l'information "input", est doté de trois champs correspondant à son nom, son type, et une étiquette sémantique (« semtag ») qui le caractérise à plus haut niveau au plan sémantique. Les trois informations qui seront ensuite utilisées lors du test de compatibilité entre services sont : l'information d'entrée ou de sortie (« input » ou « output»), l'information de type et l'étiquette sémantique.

[0031] Des explications détaillées sont maintenant données concernant l'analyse sémantique.

20 **[0032]** Lorsqu'un fichier WSDL est analysé : quatre parties principales peuvent être identifiées qui correspondent aux types, messages, opérations et nom du service. En outre, d'autres informations correspondant au « PortType » et au « Binding » sont identifiées (ces dernières terminologies étant des terminologies classiques d'une description WSDL), lesquelles serviront à faire le lien entre les entrées/sorties de l'opération et les messages.

25 **[0033]** Un premier niveau d'analyse, purement syntaxique, de la description WSDL permet d'obtenir la structure syntaxique « utile » du service, c'est-à-dire les informations nécessaires et suffisantes. On obtient ces informations principalement à partir des quatre parties précitées. Les éléments inutiles au but recherché sont enlevés. Les informations restantes sont réordonnées et inter reliées grâce aux informations de « Binding » et de « PortType ». Ceci s'obtient facilement à l'aide d'un analyseur syntaxique. A ce niveau, les étiquettes sémantiques ne sont pas encore posées.

30 **[0034]** A titre d'exemple, la structure syntaxique utile obtenue pour le service de traduction est la suivante (avec les terminologies du langage XML) :

```

  <service name="Translator">
35
      <types /> //cette entrée est vide, car il n'y a pas
                //de type complexe
40
      <operation name="Translate">
          <input>
35              <part name="lang" type="string" />
                <part name="text" type="string" />
          </input>
50
          <output>
              <part name="TranslateResult" type="string" />
55          </output>
          </operation>

```

</service>

5 **[0035]** A partir de la structure syntaxique utile, une analyse plus fine est faite à l'aide du module sémantique. Ce module peut par exemple se charger des opérations suivantes. Il repère les termes utilisés (c'est-à-dire ci-dessus les termes « Translate », « lang », « text », « TranslateResult »), les éclate (ces termes sont généralement écrits de manière similaire aux termes utilisés dans le langage de programmation ou séparés par des caractères « _ », etc.), afin de déconcaténer ces termes. Le module expande ensuite (par recherche dans un dictionnaire) les termes tronqués si nécessaire. On obtient dans l'exemple précédent : « translate », « language », « text » et « translate result ». Après cela et au moyen de dictionnaires appropriés, Il cherche à associer un nom à chaque verbe rencontré ainsi que des définitions de chaque nom. Le cas échéant, il procède à un examen de ces définitions pour en conserver une plutôt qu'une autre. Des techniques complémentaires sont connues, lesquelles permettent, le cas échéant, de simplifier ces traitements et d'aboutir à des résultats facilement exploitables.

15 **[0036]** La cinquième étape, ci-dessus, a comme principal but de poser une étiquette sémantique sur chacun des noms de chaque entrée/sortie de la structure syntaxique utile précitée du service. On posera ainsi par exemple l'étiquette "text" pour l'entrée "text". De la sorte, on obtient une description remaniée du service. Dans l'exemple du service de traduction, on obtient une description semblable à l'extrait de fichier XML décrit plus haut.

20 **[0037]** Apposer une étiquette sémantique - laquelle peut être vue comme un « super type » - permet une détermination efficace de la connectivité des services web. D'autres techniques d'analyse sémantique peuvent être envisagées pour cette apposition.

[0038] On dispose à présent de tous les éléments permettant la connexion avec un autre service.

[0039] Dans ce qui suit, l'étape de détermination de connectivité de services web est décrite au travers d'un exemple.

25 **[0040]** Dans cet exemple, on souhaite obtenir l'affichage du résultat d'une traduction d'un flux RSS (par exemple issu de pages web d'un quotidien) à partir d'un service de traduction et d'un service RSS. RSS (pour « Really Simple Syndication ») est un format connu, utilisé pour décrire le contenu d'une page Web, basé sur le XML, qui permet d'indexer de façon automatisée le contenu d'un site Web et de le mettre à disposition. Un flux RSS reprend notamment automatiquement soit les titres, soit le texte intégral d'un site d'actualité. Ce format est particulièrement utilisé pour exploiter des pages dont le contenu doit être mis à jour régulièrement. Dans le présent cas de figure, on dispose des services web nécessaires. Un problème qui se pose est donc de pouvoir composer automatiquement ces services pour obtenir une traduction automatique d'un contenu RSS.

30 **[0041]** On dispose tout d'abord d'un service web de traduction, qui possède une entrée (un texte écrit dans une première langue ou « langue 1 ») et une sortie (un autre texte écrit dans une deuxième langue ou « langue 2 »). L'analyse de la description WSDL correspondante conduit à l'identification de données d'entrée et sorties, décrites par les ensembles de triplets suivants :

35 {entrée 1, type=string, étiquette sémantique=texte à traduire} ;
 {entrée 2, type=string, étiquette sémantique=langue 1 } ;
 {sortie 1, type=string, étiquette sémantique=texte traduit} ; et
 40 {sortie 2, type=string, étiquette sémantique=langue 2}.

[0042] Le mot « string » indique une chaîne de caractères.

[0043] On dispose aussi du service web fournissant un flux RSS. Ce service web fournit par exemple, après analyse par l'outil précité, des sorties spécifiées comme suit :

45 {sortie 1, type=string, étiquette sémantique=titre} ;
 {sortie 2, type=string, étiquette sémantique=langue du titre} ;
 {sortie 3, type=string, étiquette sémantique=description} ; et
 {sortie 4, type=string, étiquette sémantique=langue de la description} ;
 50

[0044] On dispose en outre d'un service d'affichage, dont les caractéristiques identifiées après analyse peuvent se lire comme suit :

55 {entrée=texte à afficher, type=string, étiquette sémantique=texte} ; et
 {sortie=affichage, type=écran, étiquette sémantique=affichage} ;

[0045] La connectivité de chacun des services à composer est ensuite déterminée, de préférence deux à deux. Pour

EP 1 811 373 A1

ce faire, on peut rechercher une correspondance sémantique entre la ou les entrées de l'un des services web et la ou les sorties d'un autre service, c'est-à-dire une compatibilité entre deux triplets sémantiques.

[0046] Cette compatibilité peut, en particulier, répondre à trois conditions :

- 5
- a) le premier triplet concerne une sortie, l'autre une entrée
 - b) les deux types sont identiques ou compatibles
 - c) les deux étiquettes sémantiques sont identiques ou compatibles.

[0047] Une table de compatibilité pourra à cet égard être prévue et mise à disposition de l'outil informatique précité.

10 **[0048]** Dans l'exemple ci-dessus, une compatibilité est automatiquement trouvée dans le sens :

Service RSS -> service de traduction -> Service d'affichage

15 **[0049]** En effet, une table mise à disposition du module sémantique peut permettre d'établir une parfaite correspondance deux à deux, en utilisant chacun des services.

[0050] Au contraire, considérons la séquence suivante :

Service RSS -> Service d'affichage -> Service de traduction.

20 **[0051]** Cette séquence ne sera pas prise en compte du fait de l'incompatibilité sémantique entre la sortie du service d'affichage et l'entrée du service de traduction.

[0052] Notons qu'un processus itératif ou « répétitive de traitement » peut, le cas échéant, être adopté afin de coupler un nombre donné de sorties de même nature d'un premier service à une entrée d'un autre service. Ceci est particulièrement utile dans l'exemple précédent. De la sorte, chaque sortie du service RSS (titre, description, etc.) est traduite puis affichée.

25 **[0053]** Plus en détails, l'étape de détermination de la connectivité des services web permet peut être effectuée en deux temps :

- 30
- (i) d'abord selon un ordre logique issu de l'examen de leurs entrées-sorties respectives ; et
 - (ii) ensuite par la connexion des entrées-sorties elles-mêmes, selon l'examen détaillé de leurs étiquette sémantique et de leurs types (soit en effectuant le test de compatibilité évoqué plus haut).

[0054] Ainsi, un module spécifique se charge de déterminer globalement un ordre d'orchestration des services (tel service vient avant tel autre ...) puis de déterminer à un niveau détaillé quelle sortie d'un service se connecte avec quelle entrée d'un autre service.

35 **[0055]** L'ordre d'orchestration global peut, le cas échéant, être proposé à un utilisateur ou un opérateur pour validation. Celui-ci peut alors décider de changer cet ordre ou de faire intervenir un autre service.

[0056] L'ordre d'orchestration est par exemple déterminé sur la base d'heuristiques qui peuvent être implémentées dans une base de règles et placées sous le contrôle d'un moteur d'inférence. Par exemple si un service A possède en entrée-sortie des étiquettes sémantiques correspondant à du texte et un service B a en entrée des étiquettes différentes ou incompatibles, alors l'ordre sera préférentiellement : B puis A.

40 **[0057]** Dans l'exemple décrit plus haut : on commencera par l'appel du service RSS, qui a en sortie des étiquettes sémantiques correspondant à du texte (l'étiquette en entrée correspond par exemple à une adresse web et est différente de « texte »). Ensuite, on appelle le service de traduction qui, lui, a en entrée et en sortie une étiquette correspondant à du texte. Donc, selon le schéma exposé plus haut, le service de traduction est placé en aval du service RSS. En outre, la sortie du service RSS pourra être connectée à l'entrée du service de traduction.

45 **[0058]** Au vu des compatibilités manifestes issues de la comparaison des étiquettes sémantiques et du type, les propositions de connexion automatiques seront les suivantes

- 50
- la sortie texte correspondant à des titres du service RSS est susceptible d'être connectée à l'entrée (unique) texte du service de traduction ; et
 - la sortie texte correspondant à une description du service RSS est susceptible d'être connectée à l'entrée (unique) texte du service de traduction.

55 Ces propositions peuvent être directement retenues pour l'étape suivante de composition ou, le cas échéant, être validées par un utilisateur, comme indiqué plus haut.

[0059] Ensuite, une fois que la connectivité a été établie, la composition effective des services web, en fonction de leur connectivité, est réalisée par un module spécifique d'assemblage. A cet égard, une chaîne de connexions peut par

EP 1 811 373 A1

exemple être traduite dans un langage formel d'orchestration ou de chorégraphie, tel que BPEL (voir la définition plus haut). La chaîne de connexions peut alors être soumise à un moteur d'orchestration (ou "orchestration engine"), qui se chargera de l'exécuter comme s'il s'agissait d'un service web à part entière. En particulier, une chaîne de connexions du type :

5

entrée 1 d'un service A => produit sortie 1
-> entrée 2 du service B => produit sortie 2
-> entrée 3 du service C => produit résultat final 3,

10 peut être traduite puis soumise à un moteur qui l'exécutera comme un service D à part entière composé de A, B et C, et qui, à partir de l'entrée 1, peut fournir directement le résultat final 3.

[0060] Le module d'assemblage permet dès lors une connexion automatique des services web.

15 **[0061]** On obtient ainsi un service composé automatiquement par l'outil informatique précité. L'intégration de cet outil dans certaines applications pourrait permettre d'obtenir une composition de service en temps réel. Dans l'exemple ci-dessus, il est possible d'obtenir un flux RSS traduit dans une langue souhaitée, sur une requête en ce sens et précisant les éléments nécessaires. D'autres exemples de compositions peuvent naturellement être envisagés.

Revendications

20

1. Procédé de composition automatique de services web, comprenant les étapes :

25

- d'identification de caractéristiques de services web à partir d'une description standardisée de services web ;
- de détermination d'une connectivité de ces services en fonction d'au moins une caractéristique identifiée pour chacun des services ; et
- de composition de ces services en fonction de leur connectivité.

30

2. Le procédé selon la revendication 1, dans lequel les caractéristiques identifiées comprennent des étiquettes sémantiques.

3. Le procédé selon la revendication 1 ou 2, dans lequel l'étape d'identification de caractéristiques de services web comprend, pour chacun des services :

35

- l'accession à une description standardisée du service web ; et
- l'identification de caractéristiques de ce service à partir la description standardisée de ce service.

4. Le procédé selon la revendication 3, dans lequel chaque description est en langage WSDL.

40

5. Le procédé selon l'une des revendications 1 à 4, dans lequel l'étape d'identification de caractéristiques comprend une analyse syntaxique, et sémantique de la description de services web.

6. Le procédé selon l'une des revendications 1 à 5, dans lequel les caractéristiques identifiées impliquent au moins une entrée et/ou une sortie pour chacun des services.

45

7. Le procédé selon la revendication 6, dans lequel l'étape de détermination de la connectivité des services web comprend :

- une étape de test de compatibilité impliquant au moins une caractéristique identifiée pour chacun des services.

50

8. Le procédé selon la revendication 6 ou 7, dans lequel les caractéristiques comprennent, pour chacun des services web :

55

- au moins deux ensembles comprenant au moins trois champs chacun, et dans lequel :
 - un premier champ de l'un des ensembles a trait à une entrée du service;
 - un premier champ de l'autre ensemble a trait à une sortie du service;
 - un deuxième champ de chacun des ensembles a trait à un type de données relatif au premier champ ;
 - un troisième champ de chacun des ensembles a trait à une étiquette sémantique relative au premier champ,

EP 1 811 373 A1

et dans lequel l'étape de test de compatibilité implique les trois champs de chacun des ensembles.

5 9. Le procédé selon l'une quelconque des revendications 1 à 8, dans lequel l'étape de composition comprend une étape d'orchestration des services web.

10 10. Le procédé selon l'une quelconque des revendications 1 à 8, dans lequel l'étape de composition comprend une étape de chorégraphie des services web.

11. Procédé de création d'un nouveau service web comprenant les étapes du procédé selon l'une quelconque des revendications 1 à 10 et comprenant en outre une étape de :

- création d'un nouveau service web à partir de la composition des services web.

12. Produit de programme d'ordinateur adapté à mettre en oeuvre les étapes du procédé selon l'une des revendications 1 à 11.

13. Système informatique comprenant des moyens de code adaptés à mettre en oeuvre les étapes du procédé selon l'une des revendications 1 à 11.

20 **Revendications modifiées conformément à la règle 86(2) CBE.**

1. Procédé de composition automatique de services web, comprenant les étapes :

- 25
- d'identification de caractéristiques de services web à partir d'une description standardisée de services web ;
 - de détermination d'une connectivité de ces services en fonction d'au moins une caractéristique identifiée pour chacun des services ; et
 - de composition de ces services en fonction de leur connectivité ;

30 **caractérisé en ce que** l'étape d'identification de caractéristiques comprend une analyse syntaxique, et sémantique de la description de services web.

35 2. Le procédé selon la revendication 1, dans lequel les caractéristiques identifiées comprennent des étiquettes sémantiques.

3. Le procédé selon la revendication 1 ou 2, dans lequel l'étape d'identification de caractéristiques de services web comprend, pour chacun des services :

- 40
- l'accession à une description standardisée du service web ; et
 - l'identification de caractéristiques de ce service à partir la description standardisée de ce service.

4. Le procédé selon la revendication 3, dans lequel chaque description est en langage WSDL.

45 5. Le procédé selon l'une des revendications 1 à 4, dans lequel les caractéristiques identifiées impliquent au moins une entrée et/ou une sortie pour chacun des services.

6. Le procédé selon la revendication 5, dans lequel l'étape de détermination de la connectivité des services web comprend :

- 50
- une étape de test de compatibilité impliquant au moins une caractéristique identifiée pour chacun des services.

7. Le procédé selon la revendication 5 ou 6, dans lequel les caractéristiques comprennent, pour chacun des services web :

- 55
- au moins deux ensembles comprenant au moins trois champs chacun, et dans lequel :
 - un premier champ de l'un des ensembles a trait à une entrée du service;
 - un premier champ de l'autre ensemble a trait à une sortie du service;

EP 1 811 373 A1

- un deuxième champ de chacun des ensembles a trait à un type de données relatif au premier champ ;
- un troisième champ de chacun des ensembles a trait à une étiquette sémantique relative au premier champ, et dans lequel l'étape de test de compatibilité implique les trois champs de chacun des ensembles.

5 **8.** Le procédé selon l'une quelconque des revendications 1 à 7, dans lequel l'étape de composition comprend une étape d'orchestration des services web.

10 **9.** Le procédé selon l'une quelconque des revendications 1 à 7, dans lequel l'étape de composition comprend une étape de chorégraphie des services web.

10. Procédé de création d'un nouveau service web comprenant les étapes du procédé selon l'une quelconque des revendications 1 à 9 et comprenant en outre une étape de :

15 - création d'un nouveau service web à partir de la composition des services web.

11. Produit de programme d'ordinateur adapté à mettre en oeuvre les étapes du procédé selon l'une des revendications 1 à 10.

20 **12.** Système informatique comprenant des moyens de code adaptés à mettre en oeuvre les étapes du procédé selon l'une des revendications 1 à 10.

25

30

35

40

45

50

55



DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (IPC)
X	US 2003/163450 A1 (BORENSTEIN JORAM ET AL) 28 août 2003 (2003-08-28) * abrégé; figures 2,3,5,6 * * alinéas [0008], [0029], [0031], [0032], [0067], [0068], [0084], [0085], [0101], [0106], [0127], [0128] * -----	1-13	INV. G06F9/44
			DOMAINES TECHNIQUES RECHERCHES (IPC)
			G06F
3 Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche La Haye		Date d'achèvement de la recherche 28 juin 2006	Examineur Kingma, Y
CATEGORIE DES DOCUMENTS CITES		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire			

**ANNEXE AU RAPPORT DE RECHERCHE EUROPEENNE
RELATIF A LA DEMANDE DE BREVET EUROPEEN NO.**

EP 06 30 0062

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche européenne visé ci-dessus.

Lesdits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 28-06-2006
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets.

28-06-2006

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2003163450 A1	28-08-2003	US 2004093559 A1	13-05-2004

EPO FORM P 0460

Pour tout renseignement concernant cette annexe : voir Journal Officiel de l'Office européen des brevets, No.12/82